(19) Japanese Patent Office (JP) (12) **PATENT PUBLICATION** (A) (11) Patent publication number
Patent Publication Hei 7-78254
(43) Publicized date: Heisei 7 year (1995) March 20

(51)Int.Cl.[6]	ID Code	Office control number	FI			Location to show technology
G06T 7/00						
9/20						
		7459-5L	G06F	15/ 70	330 Q	
		7459-5L			335 Z	

Examination request: not requested
Number of claims: 4 OL (total 10 pages)

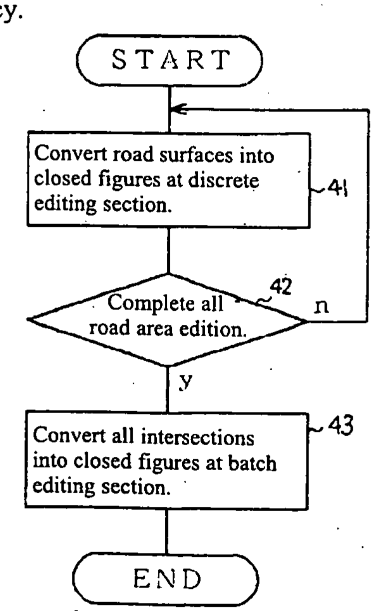
(21) Application number:
Patent Application Hei 5-223117

(22) Date of filing:
Heisei 5th year (1993) September 8

(71) Applicant: 000005108
Hitachi, Ltd.
4-6 Surugadai, Kanda, Chiyoda-ku, Tokyo
(71) Applicant: 000164438
Kyushu Electric Power Co., Ltd.
2-1-82 Watanabedori, Chuo-ku,
Fukuoka-shi, Fukuoka-prefecture
(72) Inventor: Koji Takada
Hitachi, Ltd. Information Systems Division
890-12 Kashimada, Saiwai-ku,
Kawasaki-shi, Kanagawa-prefecture
(72) Inventor: Takayuki Saito
Hitachi, Ltd. Information Systems Division
890-12 Kashimada, Saiwai-ku,
Kawasaki-shi, Kanagawa-prefecture
(74) Attorney: Katsuo Ogawa, Patent agent
Continued to the last page.

(54) **[Title of the invention]** Extraction method for closed region of figure

(57) **[Summary]**
[Objective] To cut out open region intervals as a set of closed region figures from entered regions being composed with multiple polygonal vectors.
[Constitution] In a case wherein polygonal vectors are pre-entered as outlines in a region where area recognition is conducted such as roads on a map, and when an operator selects a point within the region, it traces polygonal line vectors from the enter point and cut outs polygonal line vectors. It generates hidden line vectors on both ends of the cut out open polygonal lines, and outputs a closed figure which is surrounded with the cut out polygonal line vectors and hidden line vectors. It repeats these operations, and regions of such as intersections are considered as closed regions being composed with remaining polygonal line vectors and hidden line vectors and conducts cut out of closed figured at one time.
[Effect] It is able to cut out road sections of maps, for example, as closed figures in high efficiency.



[Claims]

[Claim 1] In graphic processing device which edits and outputs open regions of figure which are described in drawings and described with multiple polygonal line vectors, as a set of multiple close region figures; extraction method for closed region of figure which is characterized by containing following protocols of 1) through 5).

1) Select two polygonal line vectors near the point which is designated by an operator among said open regions,

2) cut out polygonal line vectors until selected two polygonal line vectors become practically not parallel, and outputs closed regions by connecting the cut out polygonal line vectors with hidden line vectors,

3) repeat above described protocols of 1) and 2),

4) for regions which were not cut out in the processing of 3), define and output closed regions by using polygonal line vectors existing in that region and hidden lined being defined by above described protocol of 2), and

5) repeat the protocol of 4) for the regions which were not cut out in the processing of 3).

[Claim 2] Extraction method for closed region of figure of Claim 1 wherein the protocol of above described 1) is to select two which are closest among polygonal line vectors which exist within a range which has been determined beforehand.

[Claim 3] Extraction method for closed region of figure of Claim 2 wherein the protocol of above described 2) is to obtain angle θ between two polygonal line vectors and to judge that they are not parallel when this θ is within a range which has been determined beforehand.

[Claim 4] Extraction method for closed region of figure of Claim 3 wherein a closed region is defined by providing hidden line vectors along the drawing region border in a case when the object region to be cut out contacts with the drawing region border, in above described 4).

[Detailed explanation of the invention]

[0001]

[Application field in industry] This invention concerns extraction method of closed region of figures in graphic processing device, and is able to apply in a case when recognizing regions such as roads on a geological map as areas (closed region, closed figure), and further, it concerns extraction method of closed region of figures which is suitable for editing such as roads on geological maps of which outline have been entered as polygonal line vectors in advance as a set of multiple closed regions.

[0002]

[Prior technology] Editing elements such as roads on geographical maps as a set of closed regions having an attribute of area is necessary to judge in which region on the geographical map the designated belongs, when an arbitrary point on a geographical map is designated. According to previous method, it has been done by a method which is to manually define and enter a closed region which is to be recognized, in order to recognize a region like above described with a graphic processing device. For example in Figure 2, when a region of roads in the geographical map 21 is to be recognized as an area, it has been done that an operator defines by splitting it into closed regions of f1 through f7 for the road area which is to be recognized as an area, as shown in enlarged illustration 22, and closed figure vector of each closed region is manually entered. On the other hand, entering outlines only as open figure vectors for such as roads on geographical maps, a method has been known wherein a geographical map is entered with a graphic entering device such as an image scanner in a graphic processing device and the outlines of such as roads on the entered image graphics are traced and converted them into vectors.

[0003]

[Problems to be solved by this invention] Above described manual entering method requires significant time in operation, and entering large quantity of graphics with this method has been very inefficient. Further, the method to turn into vector by tracing outlines is effective for cutting out of a close region which is surrounded four sides with roads (cross lined region in Figure 2, for example), however, there have been a need to manually convert necessary regions into closed region after entering only outlines as open figure vectors as described above, when an attempt is made to recognize an open region such as roads as a set of closed regions.

[0004] The objective of this invention is to provide extraction method for closed region of figure which defines open regions which do not form closed regions as a set of closed regions by using polygonal line vectors for regions such as roads in geographical map wherein outlines have been entered as polygonal line vectors.

[0005]

[Means to solve the problems] In order to accomplish above objective, this invention first classify a road into road areas a and intersections b as shown in Figure 1. This road is an open region which is not made to a closed region by polygonal line vectors when it is viewed from a point on the road. For road area a, a graphic processing device will automatically cut out polygonal line vectors (C in Figure 1) which are contained in this region when an operator simply designate one point within this region, and it generates hidden line vectors (dashed lines in enlarged section in Figure 1) on both ends and makes an output as cut out of a closed region vectors which comprises the polygonal line vectors and the hidden line vectors. For the intersection b, it can be considered as a closed region by using hidden line vectors which are generated at editing of closed region of the road area a. Accordingly, it is able to define all road areas as a set of areas (closed regions) by selecting figures from one end of open polygonal lines in sequence until reaching to the end point on the opposite side and editing selected open polygonal lines into closed region vectors.

[0006] A graphic processing device which realizes above described extraction method is equipped with a graphic data storage means, a coordinates entering means, a display means to display graphic data, and a graphic handling and processing means which handles graphic data. The graphic handling and processing means may be realized in software by a program, and it is generally separated as a discrete editing section and a batch editing section. The discrete editing section is provided with tracing object figure extraction section which extracts polygonal line vectors from the recognition initiation point that is given by an operator, a parallelism checking section which judges whether the tracing pint is included in a closed region or not, a next point searching section which finds the next control point for tracing, a tracing end point detection section which finds a end point to be a closed region, and a closed figure processing section which makes an output of a closed region which is surrounded by the cut out polygonal line vectors and hidden lined which connects end points each other. Further in the batch editing section, a processing object figure extraction section which extracts processing object figures in a drawing, a closed region composing figure extraction section which traces open line polygonal lines from an endpoint of open polygonal lines in sequence, a corner generation section which generates corners when it reaches a drawing region border, and a close figure processing section which edits the traced open figures, are provided. Each of these may be realized by software by using a program language.

[0007] By above described means, the graphic handling and processing means first

[Claims]

[Claim 1] In graphic processing device which edits and outputs open regions of figure which are described in drawings and described with multiple polygonal line vectors, as a set of multiple close region figures; extraction method for closed region of figure which is characterized by containing following protocols of 1) through 5).

1) Select two polygonal line vectors near the point which is designated by an operator among said open regions,

2) cut out polygonal line vectors until selected two polygonal line vectors become practically not parallel, and outputs closed regions by connecting the cut out polygonal line vectors with hidden line vectors,

3) repeat above described protocols of 1) and 2),

4) for regions which were not cut out in the processing of 3), define and output closed regions by using polygonal line vectors existing in that region and hidden lined being defined by above described protocol of 2), and

5) repeat the protocol of 4) for the regions which were not cut out in the processing of 3).

[Claim 2] Extraction method for closed region of figure of Claim 1 wherein the protocol of above described 1) is to select two which are closest among polygonal line vectors which exist within a range which has been determined beforehand.

[Claim 3] Extraction method for closed region of figure of Claim 2 wherein the protocol of above described 2) is to obtain angle θ between two polygonal line vectors and to judge that they are not parallel when this θ is within a range which has been determined beforehand.

[Claim 4] Extraction method for closed region of figure of Claim 3 wherein a closed region is defined by providing hidden line vectors along the drawing region border in a case when the object region to be cut out contacts with the drawing region border, in above described 4).

[Detailed explanation of the invention]

[0001]

[Application field in industry] This invention concerns extraction method of closed region of figures in graphic processing device, and is able to apply in a case when recognizing regions such as roads on a geological map as areas (closed region, closed figure), and further, it concerns extraction method of closed region of figures which is suitable for editing such as roads on geological maps of which outline have been entered as polygonal line vectors in advance as a set of multiple closed regions.

[0002]

[Prior technology] Editing elements such as roads on geographical maps as a set of closed regions having an attribute of area is necessary to judge in which region on the geographical map the designated belongs, when an arbitrary point on a geographical map is designated. According to previous method, it has been done by a method which is to manually define and enter a closed region which is to be recognized, in order to recognize a region like above described with a graphic processing device. For example in Figure 2, when a region of roads in the geographical map 21 is to be recognized as an area, it has been done that an operator defines by splitting it into closed regions of f1 through f7 for the road area which is to be recognized as an area, as shown in enlarged illustration 22, and closed figure vector of each closed region is manually entered. On the other hand, entering outlines only as open figure vectors for such as roads on geographical maps, a method has been known wherein a geographical map is entered with a graphic entering device such as an image scanner in a graphic processing device and the outlines of such as roads on the entered image graphics are traced and converted them into vectors.

[0003]

[Problems to be solved by this invention] Above described manual entering method requires significant time in operation, and entering large quantity of graphics with this method has been very inefficient. Further, the method to turn into vector by tracing outlines is effective for cutting out of a close region which is surrounded four sides with roads (cross lined region in Figure 2, for example), however, there have been a need to manually convert necessary regions into closed region after entering only outlines as open figure vectors as described above, when an attempt is made to recognize an open region such as roads as a set of closed regions.

[0004] The objective of this invention is to provide extraction method for closed region of figure which defines open regions which do not form closed regions as a set of closed regions by using polygonal line vectors for regions such as roads in geographical map wherein outlines have been entered as polygonal line vectors.

[0005]

[Means to solve the problems] In order to accomplish above objective, this invention first classify a road into road areas a and intersections b as shown in Figure 1. This road is an open region which is not made to a closed region by polygonal line vectors when it is viewed from a point on the road. For road area a, a graphic processing device will automatically cut out polygonal line vectors (C in Figure 1) which are contained in this region when an operator simply designate one point within this region, and it generates hidden line vectors (dashed lines in enlarged section in Figure 1) on both ends and makes an output as cut out of a closed region vectors which comprises the polygonal line vectors and the hidden line vectors. For the intersection b, it can be considered as a closed region by using hidden line vectors which are generated at editing of closed region of the road area a. Accordingly, it is able to define all road areas as a set of areas (closed regions) by selecting figures from one end of open polygonal lines in sequence until reaching to the end point on the opposite side and editing selected open polygonal lines into closed region vectors.

[0006] A graphic processing device which realizes above described extraction method is equipped with a graphic data storage means, a coordinates entering means, a display means to display graphic data, and a graphic handling and processing means which handles graphic data. The graphic handling and processing means may be realized in software by a program, and it is generally separated as a discrete editing section and a batch editing section. The discrete editing section is provided with tracing object figure extraction section which extracts polygonal line vectors from the recognition initiation point that is given by an operator, a parallelism checking section which judges whether the tracing pint is included in a closed region or not, a next point searching section which finds the next control point for tracing, a tracing end point detection section which finds a end point to be a closed region, and a closed figure processing section which makes an output of a closed region which is surrounded by the cut out polygonal line vectors and hidden lined which connects end points each other. Further in the batch editing section, a processing object figure extraction section which extracts processing object figures in a drawing, a closed region composing figure extraction section which traces open line polygonal lines from an endpoint of open polygonal lines in sequence, a corner generation section which generates corners when it reaches a drawing region border, and a close figure processing section which edits the traced open figures, are provided. Each of these may be realized by software by using a program language.

[0007] By above described means, the graphic handling and processing means first

conducts conversion processing to closed regions for road areas in the drawing with the discrete editing section, then after that, it conducts a batch processing for remaining regions such as intersections.

[Functions]

[0008] (1) Operation of discrete editing section

At the tracing object figure extraction section in the discrete editing section, when an operator designates a point with such as a coordinates entering device inside of a region, where he wants to edit in a region such as road wherein outline is entered as polygonal line vectors, it sets specific region from the entered point and extracts an polygonal line vector which is closest to the designated point among the figures within the specific region. Then, it extracts a polygonal line vector which is closest to the designated point out of the extracted polygonal vectors within figures wherein diagonal line is contained within specific road width in the direction of the input point. Thus, two polygonal line vectors are extracted. The parallelism checking section connects the tracing starting points and next controlling points which are in the direction of the tracing, respectively, and obtain the difference in angle of vectors, between these two polygonal line vectors. When the difference of angles is within specific parallelism margin angle, and the distance between the next control points is specific threshold width, it judges that both vectors from the tracing start points to the next control points are parallel, and it obtains coordinates information of the control points. If not judged as parallel, it returns back to the trace starting point and goes to trace in opposite direction. Then next control point of above described parallelism checking section is obtained at the next point search section. When the tracing reaches to the end point of the polygonal line vector, it searches for the coordinates of the end point and the polygonal line vector which has that coordinates and obtains the next control point. At the tracing end point detection section, it sets the position of end point as a cut surface of the closed region when it is judged not parallel to the trace result, and make an output of the part which was not traced as the result as a separate polygonal line vector. The closed figure processing section makes two polygonal line vectors which have been extracted by above sections and two lines (hidden lines) which connect end point of each of them into a group, and adds area attribute to send out. As above described, it is able to efficiently generate closed regions only by entering recognition initiation point by an operator.

[0009] (2) Operation of batch editing section

The processing object figure extraction section of the batch editing section extracts processing object figure elements in a drawing. The processing object figure elements mean open polygonal lines which are not closed region processed at discrete editing section and hidden line vectors which were generated at discrete editing section. On the closed region composing figure extraction section, it selects the figure element which was written at the very last among the figure elements being extracted at the processing object figure extracting section, and extracts a figure element having the same coordinates values with its one of the end point coordinates at an end point among the selected figure elements. It similarly conducts a search from the other end point of the extracted figure element, and it defines a figure element which is searched before reaching the other end point of the processing starting figure as the closed region composing figure element. When it reaches a drawing region border as a result of the search, it similarly traces from the other end point of the processing starting figure, and when it reaches the drawing region border again, it generates hidden line vectors on the drawing region borders. At the corner generating section, it generates hidden line vectors matching

corners of the drawing region borders, in a case when both of the x and y coordinates of two points on the drawing border are different when generating hidden line vectors in the tracing at the closed region composing figure extracting section. At the closed figure processing section, it makes the closed region composing figure element being extracted and generated hidden line vectors by above described section into a group and outputs with adding attribute of the area. At the batch editing section, it is able to efficiently generate all intersection regions in the drawing into closed figures by similarly making all the figures being extracted at the processing object figure extracting section into a closed figure group.

[0010]

[Embodiment example] One embodiment example of this invention is concretely described below by using illustrations. Although an description is made based on an example of cutting out a road on a map in this embodiment example, this invention is not limited with this. Figure 3 is a block diagram showing the structure of a graphic processing device concerning an embodiment example of this invention.

[0011] Outlines of roads which are figure elements on a map has been entered beforehand with polygonal line vectors in a data memory 3 as graphic data. Also this data memory 3 stores various data being obtained by this graphic processing device (such as data of control points and coordinates values). The graphic handling and processing section 2 comprises two programs, discrete editing section and batch editing section in general classification. The discrete editing section is equipped with tracing object figure extracting section 31, parallelism checking section 32, next point searching section 33, search end point detecting section 34 and closed figure processing section 35. The batch editing section is equipped with processing object figure extraction section 36, closed region composing figure extraction section 37, corner generation section 38 and closed figure processing section 39. Each of these sections 31 through 39 may be realized by software with programming. The central processing unit 1 conducts graphic closed region extraction processing by executing the program of the graphic handling and processing section 2. Coordinates entering device 4 is the one which is able to designate target coordinates such as a mouse or a light pen. The display unit 5 is the one which displays graphic data and it is CRT display or liquid crystal display, for example.

[0012] Next, outline of graphic closed region extraction processing is described based on process flow in Figure 4. Polygonal line vectors showing a geographical map which is to be applied with the closed figure editing are read out from the data memory 3, and portions of roads except for such as intersections and T-sections (f1, f2, f4, f6 and f7 in the case of Figure 2) are processed into closed figure (step 41) at the discrete editing section. This is repeated until the completion of this editing of closed figures for entire roads (step 42). After the completion of all, then processing into closed figure is done for those which are not able to be made into closed figures by the discrete editing section and intersection regions (f3 and f5 in the case of Figure 2) at the batch editing section (step 43), and the processing of entire road regions into closed figures is completed.

[0013] Next, detail operation of the discrete editing section is described based on the flow chart in Figure 5. When program of the discrete editing section is started, the tracing object figure extraction section 31 functions first. Figure 7 is a drawing explaining this operation. The tracing object figure extraction section 31 requests the operator for entering a recognition starting point (step 51). The operator designates coordinates which is inside of a region such as the point S in Figure 7 where he wants to extract as closed

figure by using coordinates entering device 4. The central processing unit 1 conduct the extraction of tracing object vector from the coordinates data of the entered point S (step 52). At first, it sets a rectangular region of searching width L with the point S as the center, and selects polygonal line vectors which are included within that region from the data memory 3. This searching width L should be arbitrarily selected according to the scale of the map. Then it extracts a polygonal line vector which exists closest to the entered point S among selected figures as a figure element A as shown in Figure 7(1). Then it sets a range which makes a diagonal line to be threshold road width W in a direction toward the entered point S from the figure element A, and selects polygonal line vector which is included in this range from the data memory 3 and extract as figure element B as shown in Figure 7(2). Then as shown in Figure 7(3), a tracing reference points which are for tracing the figure elements A and B are determined. These are closest points to the entered point S on the figure element A and B. Then tracing directions for the figure elements A and B are set in the same direction, and the first control points in the tracing direction from the tracing reference points are determined as tracing next points, respectively. Where, the control points are composing points on the polygonal line vectors.

[0014] The figure element A and figure element B being extracted by above described processing are cut out of control points in sequence by the parallelism checking section 32. Figure 8 is an illustration for explaining the operation of the parallelism checking section. If it is judged that a tracing next point exists in the tracing direction (step 53), a line that connects the tracing base point and the tracing next point is considered for each of the figure elements A and B and an angle θ that two lines make is obtained (step 54). At the same time, the distance between the figure elements A and B is obtained as the road width h. Following processing is done based on the relations between obtained θ and parallelism allowance angle α, and obtained h and threshold road width W.

(1) When $\theta \leq \alpha$, and $0 < W$,
it judges that the section from the tracing starting points to the tracing next points are parallel in the figure elements A and B, and respective tracing next points are registered in the data memory 3 as the closed figure composing points. Then it defines the tracing next points as the tracing base point for the next judgment and it proceeds to the step 55 and conducts search for the next points in the tracing directions (a description is made later regarding the next point searching section).

(2) When the condition of (1) is not satisfied, it judges that the section from the tracing starting points to the tracing next points are not parallel in the figure elements A and B, and conducts tracing end point detection processing in step 56 (tracing end point detection section is described later), and it registers the end points as closed figure composing points in data memory 3. If its tracing is the first time by the judgment in step 57, it reverses the tracing direction and conducts tracing processing in opposite direction.

[0015] Then, new tracing next point is obtained with the next point detection section 33 against the new tracing base point which is established by the parallelism checking section 32. Figure 9 is an illustration for explaining an example of operation of the next point searching section. In a case when the new tracing base point is not the end point of figure element as shown in Figure 9(1), a point control being proceeded toward the tracing direction from the new tracing base point is obtained, and this is defined as the tracing next point. In a case when the new tracing base point is the end point of the figure element as shown in Figure 9(2), a polygonal line vector is selected from the graphic memory within the region 91 that

contains the end point coordinates, and it extracts a figure element having the search object coordinates at the end point from selected figure. Among the extracted figure elements, the figure element 93, which is not the figure element 92 that was previously traced, is made as a new tracing object element, the end point that is at the same value with the search object coordinates is made as a new racing point, adjacent control point is made as new tracing next point, and its direction is made as the tracing direction. In a case when extracted figure element is only the figure element under tracing, tracing end point detection processing is applied, and the end point is registered in the control point registering section as the closed figure composing point. If that tracing is the first time, a tracing processing in opposite direction is conducted. Further, in a case when multiple object figures exist, it is defined as an error.

[0016] Then, a processing for generating end points of closed region is conducted (step 56) by the tracing end point detection section 34, for the tracing sections which were not judged as parallel by the parallelism checking section. Figure 10 is an explanation drawing which explains an example of graphic closed region extraction processing at the tracing end point detection section. At first, a vertical line is drawn from the tracing base point q0 of the figure element A to a straight line that connects the tracing point p0 of the figure element B and the control point p1 which is one step before it, and the cross point is defined as p2. Then following processing is applied based on positional relation of p0 and p2.

(1) In the case when distance between p0 and p2 is within W x σ ($\sigma > 1$):
Respective tracing end points of figure elements A and B are defined as tracing end points as they are.

(2) In a case when they do not satisfy (1) and p2 is not on the line p0p1:
A point that is drawn from p0 to the figure element A is used as the tracing end point of the figure element A. p0 is used as the tracing end point of the figure element B.

(3) In a case when they do not satisfy (1) and p2 is on the line p0p1:
The tracing base point of the figure element A is used as the tracing end point of the figure element A. p2 is used as the tracing end point of the figure element B. Parts which were not traced due to the generation of tracing end points are output as figure elements of open polygonal lines and previous figure elements are deleted.

[0017] Then, a processing to make the polygonal line vectors being extracted in above described processing into a group of a closed figure is done (step 59) at the closed figure processing section 35. Figure 11 is an explanation drawing explaining the figure open region extraction processing of the closed figure processing section. When the second processing at the parallelism checking section is completed and tracing points are generated, the central processing unit connects their tracing end points each other which are end points of both ends of a closed figures and generates hidden line vectors. Hidden line vectors are generated on different layer from the road vectors not to be shown on the drawing. The central processing unit makes the extracted polygonal line vectors and generated hidden line vectors into a group and outputs after applying an attribute to the "area" being surrounded with those.

[0018] Thus, regions except for the areas such as intersections and T-sections among roads are extracted as closed regions, and then extraction of such as intersection areas is conducted at the batch processing section. This operation is explained based on the flow chart on Figure 6. When the program of the batch editing section is started, the processing object figure extraction section 36 functions. In step 61, all of the polygonal line vectors and the generated hidden line vectors are extracted which have not been applied with

the closed figure editing processing by the discrete editing section among entire drawings, in step 61. Figure 12 is an explanation drawing which explains the open figure vectors being extracted at the processing object figure extraction section. It proceeds to step 62 and figure number n of the most lastly drawn among the extracting elements is memorized.

[0019] Figure 13 is an explanation drawing which explains an example of operation of the closed region composing figure extraction section 37 in figure closed region extraction processing. In step 63, P1 at the bottom side is selected as the watched point for the figure element of figure number n among the open figure vectors being extracted by the processing object figure extraction section. In step 64, the central processing unit searches for a figure which has an end point of identical coordinates with p1 except for the figure number n among the figures being extracted by the processing object figure extracting section. In step 65, the other side of the end point P3 on searched figure element is made as the watched point and a figure search is similarly conducted. It is repeated until search result reaches following conditions.

(1) The watched point returns back to the end point P2 on the other side of the search starting figure element as 131.

(2) I a case when it terminates on the drawing region border (range of error L): the other side of end point P2 is made as the watched point in step 66 and similarly conduct a figure search. And in a case when the search result reaches the drawing region border again as 132, those two points on the drawing region border are connected with a hidden line vector in step 67. In a case when x and y coordinates of the two points are both different as 133, corner generation is done in step 68. (The corner generation section is described later.)

(3) In a case when search of figure element become unavailable in conditions other than (2), it is defined as error.

[0020] Figure 14 is an explanation drawing explaining the corner generation section 38. At first, it examines on which edge do those two points on the drawing borders exist. In a case when those two points are on adjacent edges (141), polygonal line vectors having the corner which is made by two edges of the drawing region border as control point are generated as hidden line vectors. In a case when those tow points are on opposing edges (142), two corners on the drawing region border are made as control points and polygonal lines which are on shorter distance side are made as hidden line vectors.

[0021] It proceeds to step 69 and the central processing unit makes the extracted closed region composing the figures being extracted in above described processing and hidden line vectors into a group at the closed figure processing section 39, and outputs by adding attribute of "area". In step 70, the open figures which have been turned into a closed figure group is deleted from a group pf figures which have been extracted by the processing object figure extraction section 36. Further in step 71, it defines figure number of the one, which is drawn most lastly among the remaining figure elements, as n.

[0022] The batch processing section conducts above described processing, closed figure editing, until figure elements runs out which has been extracted by the processing object figure extraction section.

[0023] Although this invention was concretely described based on the embodiment example, it is needless to say that this invention is not restricted within said embodiment examples and able to be variously modified within a range which does not deviate from its outline.

[0024]

[Effect of the invention] According to this invention as described above, it is able to enter only border lines in polygonal line vectors in advance and conduct closed figure

editing to those vectors in extraction processing of closed region of figure. According to this method, it is able to efficiently conduct compared to previous method wherein closed figures must be entered as areas where are desired to be recognized as closed regions. Especially, it is an effective method when conducting processing of large quantity of drawings and large area.

[Brief explanation of drawings]

[Figure 1] An explanation drawing for road areas and intersection area which is an example of the processing object of this invention.

[Figure 2] An explanation drawing of previous technology.

[Figure 3] A block diagram showing the structure of a graphic processing device concerning an embodiment example of this invention.

[Figure 4] A flow chart showing entire processing of the embodiment example.

[Figure 5] A flow chart showing the operation of the discrete editing section.

[Figure 6] A flow chart showing the operation of the batch editing section.

[Figure 7] An explanation drawing showing an example of tracing object figure extraction processing.

[Figure 8] A drawing explaining operation of parallelism checking section.

[Figure 9] A drawing explaining operation of next point search processing.

[Figure 10] A drawing explaining operation of tracing end point detection processing.

[Figure 11] A drawing explaining operation of closed figure processing

[Figure 12] An explanation drawing of an open figure which is extracted at processing object figure extraction section.

[Figure 13] A drawing showing an extraction example of a closed region composing figure.

[Figure 14] A drawing showing operation of corner generation section.

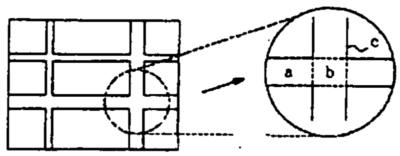
[Brief description of numbers]

1: central processing unit, 2: graphic handling and processing section, 3: data memory, 4: coordinates entering device, 5: display unit

Translated by: Hideyo Sugimura, 651-490-0233, hsugimura@pipeline.com, July 24, 2005

[Figure 1]

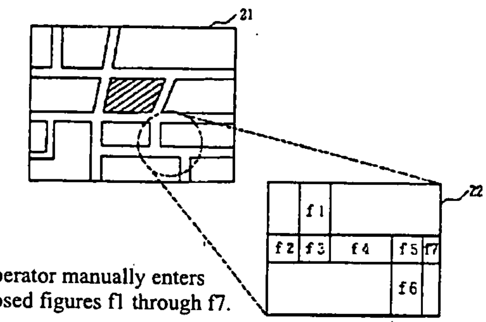
Figure 1 Road areas and intersections



a: Road area
b: intersection
c: road outline open figure vector

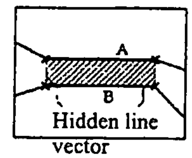
[Figure 2]

Figure 2 Previous technology



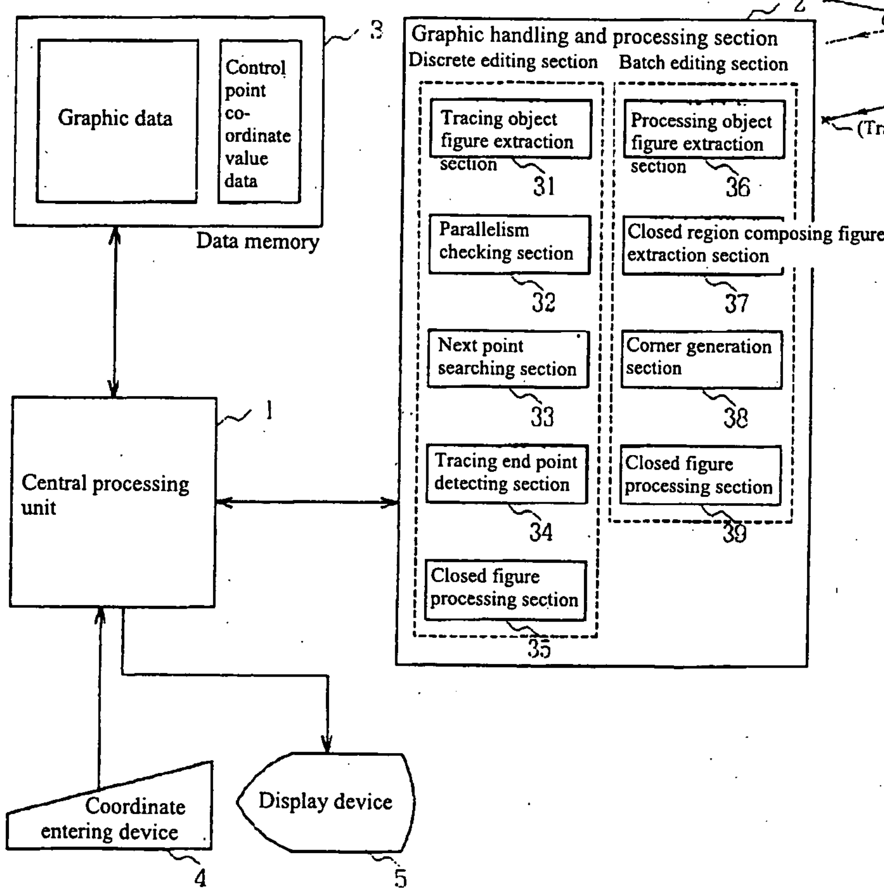
[Figure 11]

Figure 11



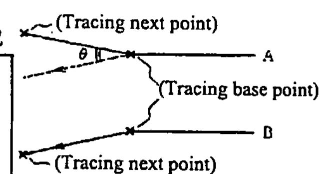
[Figure 3]

Figure 3



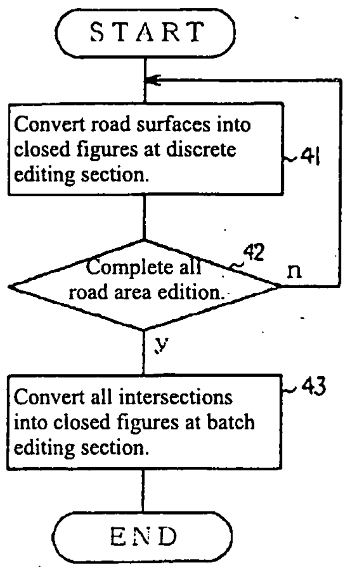
[Figure 8]

Figure 8



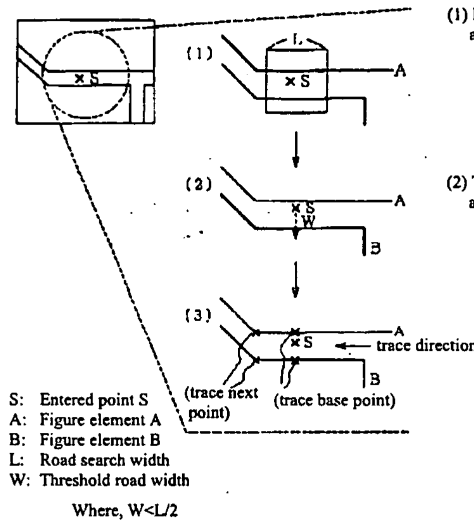
[Figure 4]

Figure 4



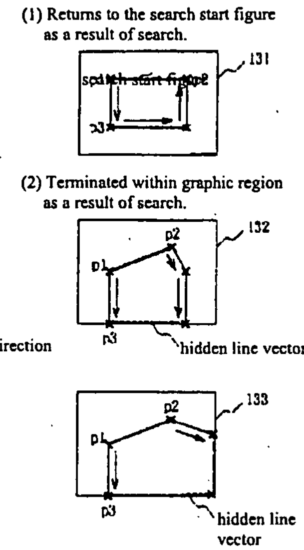
[Figure 7]

Figure 7



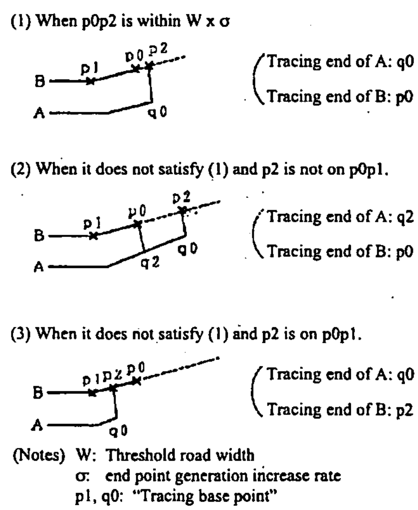
[Figure 13]

Figure 13



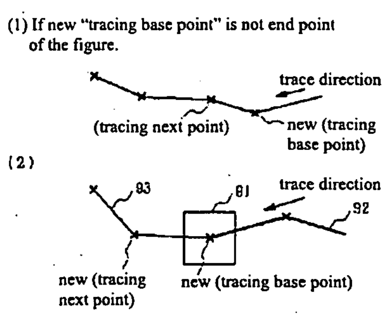
[Figure 10]

Figure 10



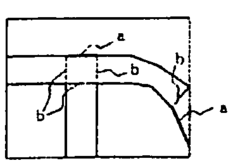
[Figure 9]

Figure 9



[Figure 12]

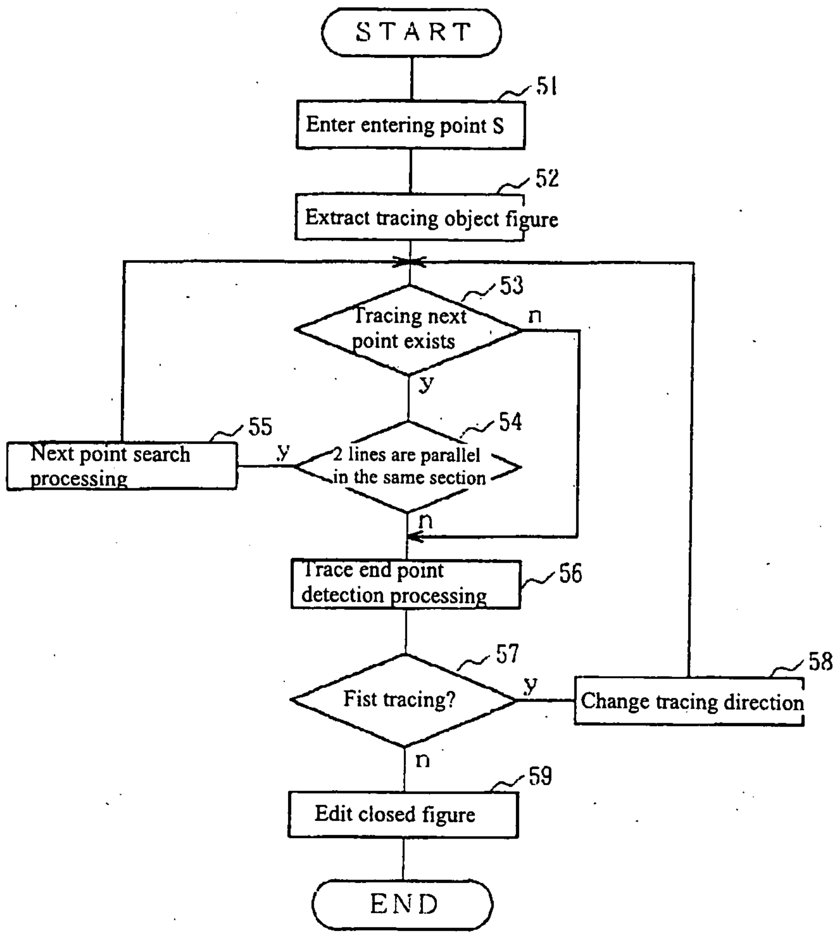
Figure 12



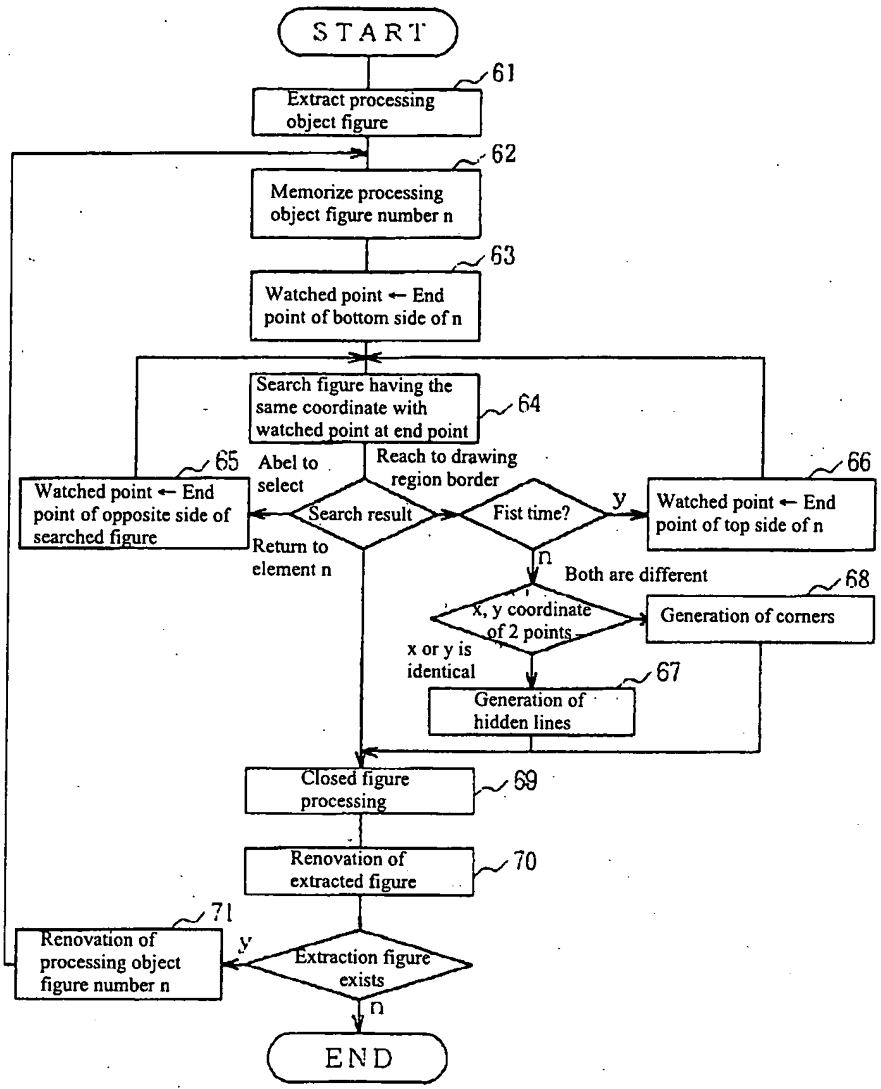
◎ Figures to be extracted
a: Those which were not converted into closed figures in discrete processing.
b: Hidden line vectors.

[Figure 5]

Figure 5



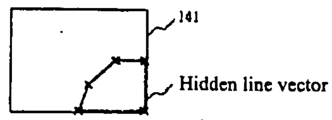
[Figure 6]


START
61
Extract processing object figure
62
Memorize processing object figure number n
63
Watched point ← End point of bottom side of n
Search figure having the same coordinate with watched point at end point
64
65
Abel to select
Reach to drawing region border
66
Watched point ← End point of opposite side of searched figure
Search result
Fist time?
y
Watched point ← End point of top side of n
Return to element n
n
Both are different
68
x, y coordinate of 2 points
Generation of corners
x or y is identical
67
Generation of hidden lines
Closed figure processing
69
Renovation of extracted figure
70
71
Renovation of processing object figure number n
y
Extraction figure exists
n
END

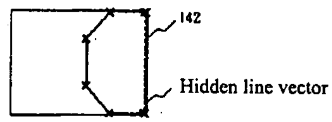
[Figure 14]

Figure 14

(1) When end points are on agjecent edges of drawing region border.



(2) When end points are on opposing edges of drawing region border.



Continued from the front page.

(72) Inventor: Masahiro Ooba
Hitachi, Ltd. Information Systems Division
890-12 Kashimada, Saiwai-ku,
Kawasaki-shi, Kanagawa-prefecture

(72) Inventor: Hiroyuki Suenaga
Kyushu Electric Power Co., Ltd.
2-1-82 Watanabedori, Chuo-ku,
Fukuoka-shi, Fukuoka-prefecture

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報(A)　(11)特許出願公開番号

特開平7－78254

(43)公開日　平成７年(1995)３月20日

(51)Int.Cl.⁶		識別記号	庁内整理番号	FI		技術表示箇所
G06T	7/00					
	9/20					
			7459－5L	G06F	15/70	330 Q
			7459－5L			335 Z

審査請求　未請求　請求項の数４　ＯＬ　（全 10 頁）

(21)出願番号　特願平5－223117

(22)出願日　平成５年(1993)９月８日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(71)出願人　000164438
九州電力株式会社
福岡県福岡市中央区渡辺通２丁目１番82号

(72)発明者　高田　浩二
神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報システム事業部内

(72)発明者　斉藤　孝幸
神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報システム事業部内

(74)代理人　弁理士　小川　勝男

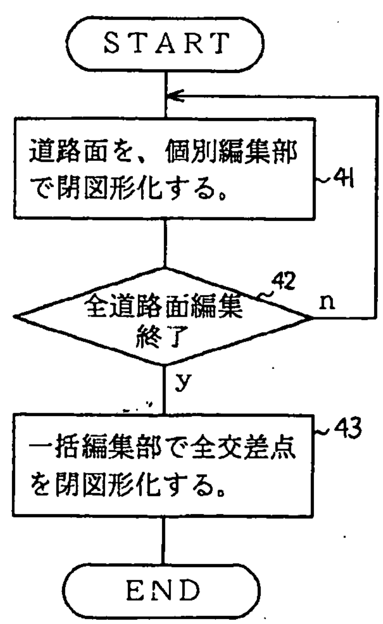
最終頁に続く

(54)【発明の名称】　図形閉領域抽出方法

(57)【要約】
ɑÇ■♥0☺º.ü:】入力済みの複数の折線ベクトルから構成される
ら、閉領域区間を閉領域図形の集合として切り出

】地形図上の道路など、面認識を行う領域にあら
輪郭線として折線ベクトルが入力されている場合、操作者が領域の内側の点を指示すると、入力点から折線ベクトルを追跡し、折線ベクトルを切り出す。切り出した開折線の両端に隠線ベクトルを生成し、切り出した折線ベクトルと隠線ベクトルによって囲まれる閉図形を出力する。これらの動作をくり返し、交差点等の領域は、残った折線ベクトルや隠線ベクトルで構成されている閉領域として一括して閉図形の切り出しを行う。
【効果】例えば、地図の道路部分を効率良く閉図形化して切り出す事ができる。

図4



【特許請求の範囲】

【請求項１】図面に記載された複数の折線ベクトルによって表現された図形の開領域を、複数の閉領域の図形の集合として編集して出力する図形処理装置において、以下の1)から5)までの手順を含む事を特徴とする図形閉領域抽出方法

1) 前記開領域のうち、操作者によって指示された点付近の２本の折線ベクトルを選択し、

2) 選択された２本の折線ベクトルが実質的に平行でなくなるまで折線ベクトルを切り出し、切りだされた折線ベクトルを隠線ベクトルでつなぐ事によって閉領域を出力し、

3) 前記1)と2)の手順をくり返し、

4) 3)の処理によって切り出さなかった領域については、その領域にある折線ベクトルと、上記2)の手順で定義した隠線を用いて閉領域を定義して出力し、

5) 3)の処理によって切り出さなかったそれぞれの領域について、4)の手順を繰り返す。

【請求項２】前記1)の手順は、指示された点を含むあらかじめ定められた範囲内に存在する折線ベクトルのうちもっとも近くにある２本を選択する請求項１の図形閉領域抽出方法。

【請求項３】前記2)の手順は、２本の折線ベクトルのなす角θを求め、このθがあらかじめ定められた範囲内にある場合に平行でないとする請求項２の図形閉領域抽出方法。

【請求項４】前記4)において、切り出し対象の領域が図面領域枠に接する場合、該図面領域枠に沿って隠線ベクトルを設けて閉領域を定義する請求項３の図形閉領域抽出方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、図形処理装置における図形閉領域抽出方法に関し、例えば地形図上の道路のような領域を面（閉領域、閉図形）として認識する場合に適用でき、既に輪郭線が折線ベクトルで入力されている地形図上の道路等を複数の閉領域の集合として編集するのに好適な図形閉領域抽出方法に関する。

【０００２】

【従来の技術】地形図上の道路等の要素を面の属性を持つ閉領域の集合として編集することは、地形図上の任意の地点を指定した際に、指定点が地形図上のどの領域に含まれているかを判定するために必要である。従来の方法によれば、図形処理装置において上記のような領域の認識を行うために、認識しようとする領域に対して人手により閉領域を定義して入力する方法で行っていた。例えば図２において、地形図21上の道路領域を面として認識しようとする場合、拡大図22に示すように、面認識しようとする道路面に対し、操作者がｆ１～ｆ７の閉領域に分割して定義し、各閉領域の閉図形ベクトルを手入力していた。一方、地形図上の道路等に対して、輪郭線のみを開図形ベクトルで入力することについては、図形処理装置においてイメージスキャナ等の図面入力装置で地形図を入力し、入力した画像図面上の道路等の輪郭線を追跡してベクトル化する方法が知られている。

【０００３】

【発明が解決しようとする課題】上記のような人手による入力方法では、操作にかなりの時間を要し、大量の図面に対してこの方法で入力することは非常に効率の悪いものであった。また、輪郭線を追跡してベクトル化する方法は、地図の中でも四方を道路などで囲まれた閉領域（例えば、図２の斜線部）の切り出しには有効であるが、道路等の開領域を、閉領域の集合として認識しようとする場合、上記のように輪郭線のみを開図形ベクトルとして入力した後に、必要な領域を人手により閉領域に変換する必要があった。

【０００４】本発明の目的は、既に折線ベクトルとして輪郭線が入力されている地形図上の道路等の領域に対し、折線ベクトルを使って閉じた領域を形成しない開領域を閉領域の集合として定義する図形閉領域抽出方法を提供することである。

【０００５】

【課題を解決するための手段】上記の目的を達成するため、本願発明は図１のように、まず道路を道路面ａと交差点ｂに分類する。この道路は、道路上の点からみると、折線ベクトルによって閉じた領域となっていない開領域である。道路面ａに関しては、操作者がこの領域の内側の一点を指示するだけで、図形処理装置が自動的にこの領域に含まれる折線ベクトル（図１のＣ）を切り出し、両端には隠線ベクトル（図１の拡大部の点線）を生成し、折線ベクトルと隠線ベクトルをからなる閉領域ベクトルとして切りだして出力する。交差点ｂに関しては、道路面ａの閉領域編集時に発生する隠線ベクトルを用いて閉領域とみなすことができる。従って、開折線の一方の端点から順次図形を選択し、反対側の端点に達するまでに選択された開折線を閉領域ベクトルに編集することにより、すべての道路面を面（閉領域）の集合として定義する事ができる。

【０００６】上記抽出方法を実現する図形処理装置は、図形データ格納手段と、座標入力手段と、図形データを表示する表示手段と、図形データの操作を行う図形操作処理手段とを備える。図形操作処理手段はプログラムによってソフト的に実現でき、大きく分けて、個別編集部と一括編集部に分かれる。個別編集部には、操作者の指示した認識開始点から折線ベクトルを抽出する追跡対象図形抽出部と、追跡点が閉領域に含まれるか判定する平行判定部と、追跡する次の制御点を求める次点検索部と、閉領域となる端点を求める追跡終点検出部と、切り出された折線ベクトルと端点同士を結んだ隠線で囲まれる閉領域を出力する閉図形処理部とを備える。また一括

編集部には、図面中の処理対象図形を抽出する処理対象図形抽出部と、開折線の端点から順次開折線を追跡していく閉領域構成図形抽出部と、図面領域枠に達した場合に角を生成する角生成部と、追跡された開図形を編集する閉図形処理部を設ける。これらの各部もプログラム言語を用いてソフトウェア的に実現する事ができる。

【０００７】

【作用】上記手段により、図形操作処理手段は、まず図面上の道路面に対しては個別編集部によって閉領域化処理を行い、その後に残りの交差点等の領域に対して一括編集を行う。

【０００８】(1)個別編集部の動作

個別編集部の追跡対象図形抽出部では、折線ベクトルとして輪郭線が入力されている道路等の領域に対し、操作者が編集したい領域の内側を座標入力デバイス等で指示すると、該入力点より所定領域を設定し、該所定領域内の図形の中で指定位置から最も近い折線ベクトルを抽出する。次に、取り出された折線ベクトルより、該入力点方向に対角線が所定道路幅以内に含まれる図形の中で、最も指定位置に近い折線ベクトルを抽出する。このようにして２つの折線ベクトルが抽出される。平行判定部はこれら２つの折線ベクトルにおいて、それぞれの追跡開始点と追跡方向上にある次の制御点とを結び、両ベクトルの角度差を求める。その角度差が所定平行余裕角度以下であり、また次制御点間の距離が所定限界幅以下の場合、追跡開始点から次制御点までの両ベクトルは平行であると判定し、制御点の座標情報を取得する。平行と判定されない場合は、追跡開始点に戻り反対方向の追跡に移る。次点検索部で上記の平行判定部の次制御点を求める。追跡が折線ベクトルの端点に達した際には、その端点座標と同座標を持つ折線ベクトルを検索し、次の制御点を求める。追跡終点検出部では、追跡の結果に平行でないと判定された場合、閉領域の切断面としての端点の位置を設定し、その結果未追跡となった部分を別折線ベクトルとして出力する。閉図形処理部では、以上の部により取り出された２つの折線ベクトルと、それぞれの端点を結んだ２線分（隠線）とをグループ化し、面属性を付与して出力する。以上のように個別編集部では、操作者が認識開始点を入力するだけで閉領域を効率良く生成することができる。

【０００９】(2)一括編集部の動作

一括編集部の処理対象図形抽出部では、図面中の処理対象図形要素を抽出する。処理対象図形要素とは、個別編集部で閉領域処理されなかった開折線、個別編集部で発生した隠線ベクトルのことである。閉領域構成図形抽出部では、処理対象図形抽出部で抽出された図形要素の中で、最も最後に描かれた図形要素に注目し、その一方の端点座標と同座標値を端点に持つ図形要素を選択図形要素の中から抽出する。抽出された図形要素のもう一方の端点から同様に検索を行い、処理開始図形のもう一方の端点に達するまでに検索された図形要素を閉領域構成図形要素とする。検索の結果図面領域枠に達した場合は、処理開始図形のもう一方の端点から同様に追跡し、再び図面領域枠に達した場合は図面領域枠上に隠線ベクトルを生成する。角生成部では、閉領域構成図形抽出部における追跡で隠線ベクトルを生成する際に、図面領域枠上の２点のｘ，ｙ座標が共に異なる場合に、図面領域枠の角に合わせた隠線ベクトルを生成する。閉図形処理部では、上記の部により抽出された閉領域構成図形要素と、生成された隠線ベクトルとをグループ化し、面の属性を付加して出力する。一括編集部では、処理対象図形抽出部で抽出された全ての図形について同様に閉図形グループ化を行うことで、図面内の全ての交差点領域を効率良く閉図形に生成することができる。

【００１０】

【実施例】以下、本発明の一実施例を図面を用いて具体的に説明する。本実施例では、地図上の道路を切り出す例をもとに説明するが、本願発明はこれに限定されるものではない。図３は本発明の一実施例にかかる図形処理装置の構成を示すブロック図である。

【００１１】データメモリ３には、地図上の図形要素である道路の輪郭線があらかじめ図形データとして折線ベクトルで入力されている。また、このデータメモリ３には本図形処理装置で求める各種データ（制御点、座標値データなど）を格納する。図形操作処理部２は、大きく分けて個別編集部と一括編集部の２つのプログラムが含まれる。個別編集部には、追跡対象図形抽出部31、平行判定部32、次点検索部33、追跡終点検出部34、閉図形処理部35を備える。一括編集部には処理対象図形抽出部36、閉領域構成図形抽出部37、角生成部38、閉図形処理部39を備える。これら各部31～39はプログラムによってソフト的に実現することができる。中央処理装置１は、図形操作処理部２のプログラムを実行して図形閉領域抽出処理を行う。座標入力デバイス４は、例えばマウスやライトペンなどのように目的とする座標を指定できるものである。表示装置５は図形データを表示するもので、例えばＣＲＴディスプレイや液晶ディスプレイである。

【００１２】次に図４の処理フローに基づき、図形閉領域抽出処理の概要を説明する。閉図形編集を行おうとする地形図を示す折線ベクトルをデータメモリ３から読みだし、道路のうち交差点やＴ字路等を除く部分（図２でいうとｆ１、ｆ２、ｆ４、ｆ６、ｆ７）を、個別編集部で閉図形化処理する(ステップ41)。全道路についてこの閉図形の編集が終わるまでこれを繰り返す(ステップ42)。全部終了したら、つぎに個別編集部によって閉図形化できないもの及び交差点領域（図２でいうとｆ３、ｆ５）については、一括編集部により閉図形化処理を行い(ステップ43)、全道路領域の閉図形化処理が終了する。

【００１３】次に個別編集部の詳細な動作を図５のフローチャートに基づいて説明する。個別編集部のプログラ

ムを起動すると、まず追跡対象図形抽出部31が動作する。図７はこの動作を説明する図である。追跡対象図形抽出部31は、操作者に対して認識開始点の入力要求を行う(ステップ51)。操作者は座標入力デバイス４を用い、図７のＳ点のように閉図形として抽出したい領域の内側を示す座標を指定する。中央処理装置１は、入力点Ｓの座標データから追跡対象ベクトル抽出を行う(ステップ52)。まず、入力点Ｓを中心に、検索幅Ｌの矩形領域を設定し、その領域内に含まれる折線ベクトルをデータメモリ３から選択する。この検索幅Ｌは地図の縮尺に応じて任意に定めれば良い。次に、選択された図形の中から、図７(1)に示すように入力点Ｓの最も近くに存在する折線ベクトルを図形要素Ａとして抽出する。次に、図形要素Ａから入力点Ｓの方向に、対角線が限界道路幅Ｗとなる範囲を設定し、この範囲に含まれる折線ベクトルをデータメモリ３から選択し、図７(2)に示すように図形要素Ｂとして抽出する。次に図７(3)に示すように図形要素Ａ、Ｂの追跡をするための追跡基準点を定める。これは図形要素Ａ、Ｂ上で入力点Ｓに最も近い点とする。次に図形要素Ａ、Ｂについて追跡方向を同一に設定し、追跡基点から追跡方向上の最初の制御点をそれぞれ追跡次点とする。ここで制御点とは、折線ベクトルの構成点である。

【００１４】上記の処理により抽出された図形要素Ａ、図形要素Ｂは、平行判定部32により制御点を順次切り出していく。第８図は、平行判定部の動作を説明するための図である。追跡方向に追跡次点が存在すると判定されたとき(ステップ53)、図形要素Ａ、Ｂそれぞれについて、追跡基点と追跡次点を結ぶ線分を考え、２本の線分のなす角θを求める(ステップ54)。同時に、図形要素Ａ、Ｂ間の距離を道路幅ｈとして求める。求めたθと平行余裕角α、及び求めたｈと限界道路幅Ｗとの関係から、以下のように処理を行う。

(1)　$\theta \leqq \alpha$　かつ　$0 < h < W$の場合

図形要素Ａ、Ｂにおいて、追跡基点から追跡次点までの区間が平行であると判定し、それぞれの追跡次点を閉図形構成点としてデータメモリ３に登録する。そして、追跡次点を次の判定時の追跡基点とし、ステップ55に進み追跡方向に次点検索を行う（次点検索部については後述する）。

(2)　(1)の条件を満たさない場合

図形要素Ａ、Ｂにおいて、追跡基点から追跡次点までの区間が平行でないと判定し、ステップ56において追跡終点検出処理を行い（追跡終点検出部については後述する。）、終点を閉図形構成点としてデータメモリ３に登録する。ステップ57の判定でその追跡が一回目であれば、追跡方向を逆にし、逆方向の追跡処理を行う(ステップ58)。

【００１５】次に次点検出部33によって、平行判定部32で設定された新たな追跡基点に対して、新しい追跡次点を求める。図９は、次点検索部33の動作の一例を説明する図である。新しい追跡基点が図９(1)に示すように図形要素の端点でない場合、新しい追跡基点から追跡方向に進めた制御点を求め、これを追跡次点とする。新しい追跡基点が図９(2)に示すように図形要素の端点である場合、端点座標を含む領域91で図形メモリから折線ベクトルを選択し、選択された図形の中から検索対象座標を端点に持つ図形要素を抽出する。抽出された図形要素の中で、前回追跡した図形要素92でない図形要素93を新しい追跡対象図形要素とし、検索対象座標と同値である端点を新しい追跡基点、隣の制御点を新しい追跡次点とし、その方向を追跡方向とする。抽出された図形要素が追跡中の図形要素のみであった場合は、追跡終点検出処理を行い、終点を閉図形構成点として制御点登録部に登録する。その追跡が一回目であれば、逆方向の追跡処理を行う。なお、対象図形が複数存在する場合はエラーとする。

【００１６】次に、追跡終点検出部34によって、平行判定部で平行であると判定されなかった追跡区間に対して、閉領域の終点を生成する処理を行う(ステップ56)。図１０は、図形閉領域抽出処理の追跡終点抽出部の一例を説明する説明図である。まず、図形要素Ａの追跡基点ｑ０より、図形要素Ｂの追跡基点ｐ０とその１つ前の制御点ｐ１を結ぶ直線に垂線を降ろし、交点をｐ２とする。そしてｐ０とｐ２の位置関係により以下の処理を行う。

(1)　ｐ０とｐ２の距離が$W \times \sigma$以内の場合（$\sigma > 1$）

図形要素Ａ、Ｂそれぞれの追跡基点をそのまま追跡終点とする。

(2)　(1)を満たさず、かつｐ２が線分ｐ０ｐ１上にない場合

図形要素Ａの追跡終点は、ｐ０から図形要素Ａに降ろした点を用いる。図形要素Ｂの追跡終点は、ｐ０を用いる。

(3)　(1)を満たさず、かつｐ２が線分ｐ０ｐ１上にある場合

図形要素Ａの追跡終点は、図形要素Ａの追跡基点を用いる。図形要素Ｂの追跡終点はｐ２を用いる。追跡終点の生成のため未追跡となった部分は、開折線の図形要素として出力し、元の図形要素を消去する。

【００１７】次に閉図形処理部35により、上記の処理で抽出された折線ベクトルを、閉図形にグループ化する処理を行う(ステップ59)。図11は、図形閉領域抽出処理の閉図形処理部の一例を説明する説明図である。２度目の平行判定部の処理が終了し追跡終点が生成されると、中央処理装置は閉領域の両側の端点となっている追跡終点をそれぞれ結びあわせ、隠線ベクトルを生成する。隠線ベクトルは、道路ベクトルとは別の板面に生成され、図面上に表示されないようにする。中央処理装置は、抽出した折線ベクトルと、生成した隠線ベクトルとをグルー

プ化し、これらによって囲まれる「面」に属性を付加して出力する。

【0018】以上のようにして、道路のうち交差点やT字路等を除く部分を閉領域として抽出するわけであるが、次に一括編集部にて交差点部など抽出を行う。この動作を図6のフローチャートに基づいて説明する。一括編集部のプログラムを起動させると、まず処理対象図形抽出部36が動作する。ステップ61において、図面全体の中で個別編集部によって閉図形編集処理がなされなかった折線ベクトル、および生成された隠線ベクトルを全て抽出する。図12は、処理対象図形抽出部によって抽出される開図形ベクトルについて説明する説明図である。ステップ62に進み、抽出する図形要素の中で、最も最後に描かれたものの図形番号nを記憶しておく。

【0019】図13は、図形閉領域抽出処理の閉領域構成図形抽出部37の動作の一例を説明する説明図である。ステップ63において、処理対象図形抽出部によって抽出された開図形ベクトルのうち、図形番号nの図形要素について、下側の端点P1を注目点とする。ステップ64で中央処理装置は、処理対象図形抽出部によって抽出された図形のうち、図形番号n以外でp1と同座標の端点を持つ図形を検索する。検索された図形要素に対し、ステップ65においてもう一方の端点P3を注目点とし、同様に図形検索を行う。検索結果が以下の状態になるまで繰り返す。

(1) 131のように、注目点が検索開始図形要素のもう一方の端点P2に戻る。

(2)図面領域枠上で途切れる場合（誤差範囲L）、ステップ66で注目点をもう一方の端点P2とし、同様に図形検索を行う。そして 132のように、再び検索結果が図面領域枠上に達した場合は、ステップ67において図面領域枠上の2点を隠線ベクトルで結ぶ。その際、133のように、2点のx，y座標が共に異なる場合は、ステップ68において角生成を行う。（角生成部については後述する。）

(3)、(2)以外の状態で図形要素の検索が行えなくなった場合はエラーとする。

【0020】図14は、角生成部38について説明した説明図である。まず、図面領域枠上の2点がどの辺上にあるか調べる。2点が隣合う辺上にある場合(141)は、図面領域枠の2辺によってなされる角を制御点とする折線を、隠線ベクトルとして生成する。2点が向かいあう辺上にある場合（142）は、図面領域枠の2角を制御点とし、距離が短くなる方の折線を隠線ベクトルとする。

【0021】ステップ69に進み中央処理装置は、閉図形処理部39によって、上記の処理において抽出された閉領域構成図形、および隠線ベクトルをグループ化し、「面」の属性を付加して出力する。ステップ70において、閉図形グループ化された開図形を、処理対象図形抽出部36によって抽出された図形群から消去する。さらにステップ71において、残りの図形要素中、最も最後に描かれたものの図形番号をnとする。

【0022】一括処理部は、以上のような処理を、処理対象図形抽出部によって抽出された図形要素が無くなるまで、閉図形編集を行う。

【0023】以上、本発明を実施例に基づき、具体的に説明したが、本発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【0024】

【発明の効果】以上説明したように本発明によれば、図形閉領域抽出処理において、あらかじめ境界線のみを折線ベクトルで入力しておき、そのベクトルに対して閉図形編集を行うことが可能である。この方法によれば、閉領域として認識したい場所に初めから閉図形を入力しなければならない従来方法に比べて効率良く行うことができる。特に、大量の図面及び広範囲に渡って処理を行う場合に有効な手段となる。

【図面の簡単な説明】

【図1】本発明の処理対象の一例である道路面と交差点面の説明図。

【図2】従来技術の説明図。

【図3】本発明の実施例にかかる図形処理装置の構成を示すブロック図。

【図4】実施例の処理全体を示すフローチャート。

【図5】個別編集部の動作を説明するフローチャート。

【図6】一括編集部の動作を説明するフローチャート。

【図7】追跡対象図形抽出処理の一例を示す説明図。

【図8】平行判定部の動作を説明する図。

【図9】次点検索処理の動作を説明する図。

【図10】追跡終点検出処理の動作を説明する図。

【図11】閉図形処理の動作を説明する図。

【図12】処理対象図形抽出部で抽出される開図形の説明図。

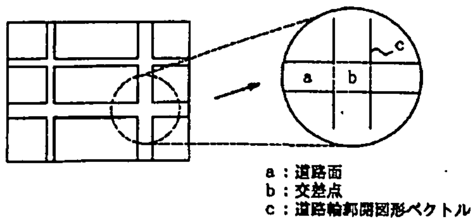
【図13】閉領域構成図形の抽出例を示す図。

【図14】角生成部の動作を示す図。

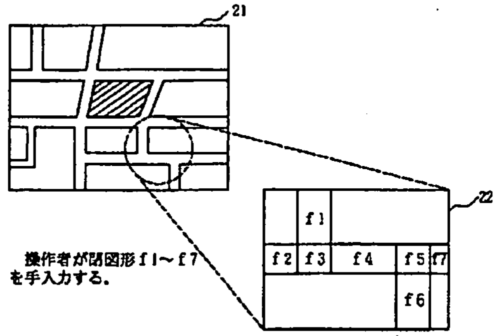
【符号の簡単な説明】

1…中央処理装置、2…図形編集処理部、3…データメモリ、4…座標入力デバイス、5…表示装置。

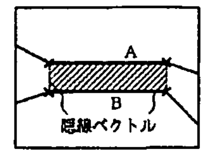
【図１】



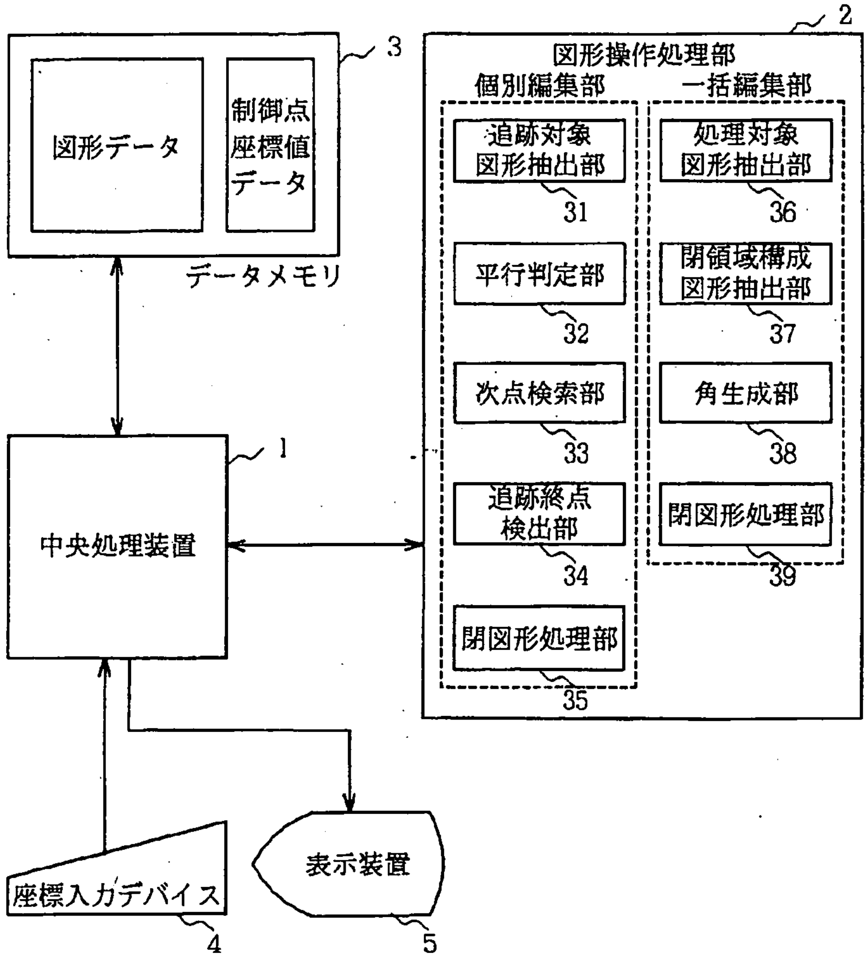
【図２】



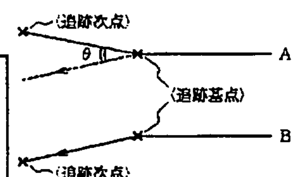
【図１１】



【図３】

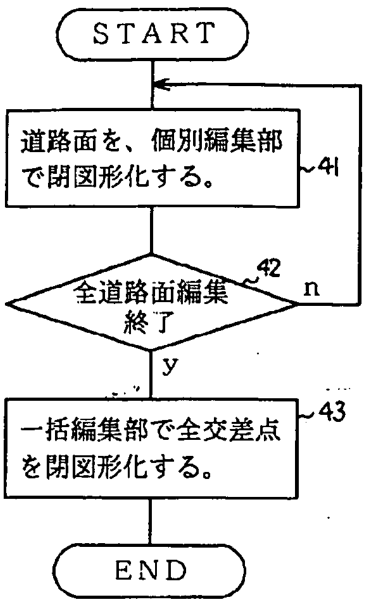


【図８】



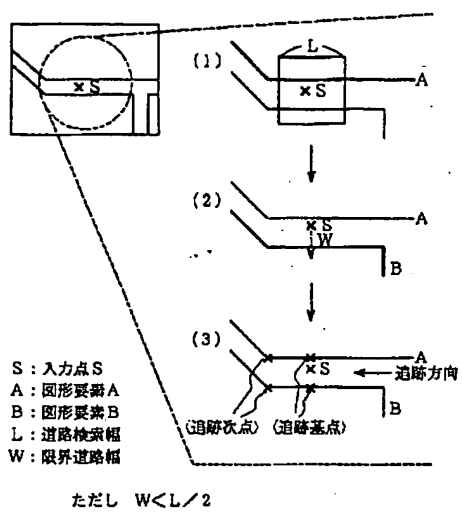
【図4】

図4



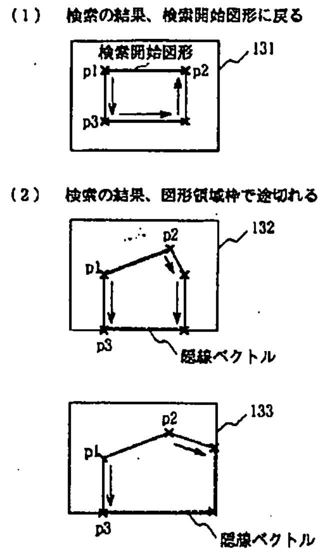
【図7】

図7



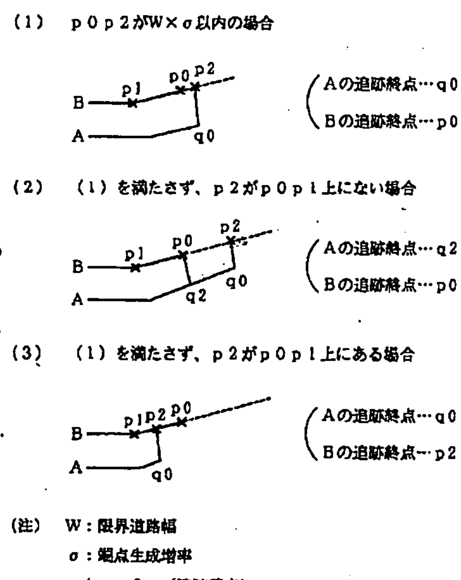
【図13】

図13



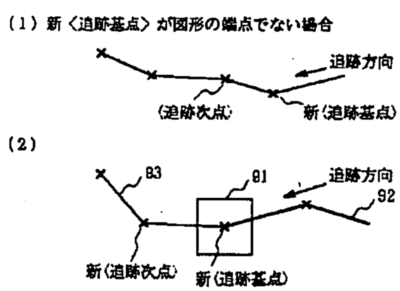
【図10】

図10



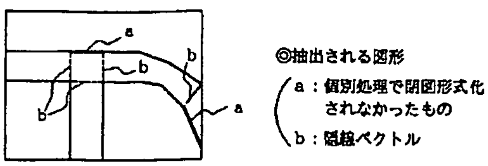
【図9】

図9



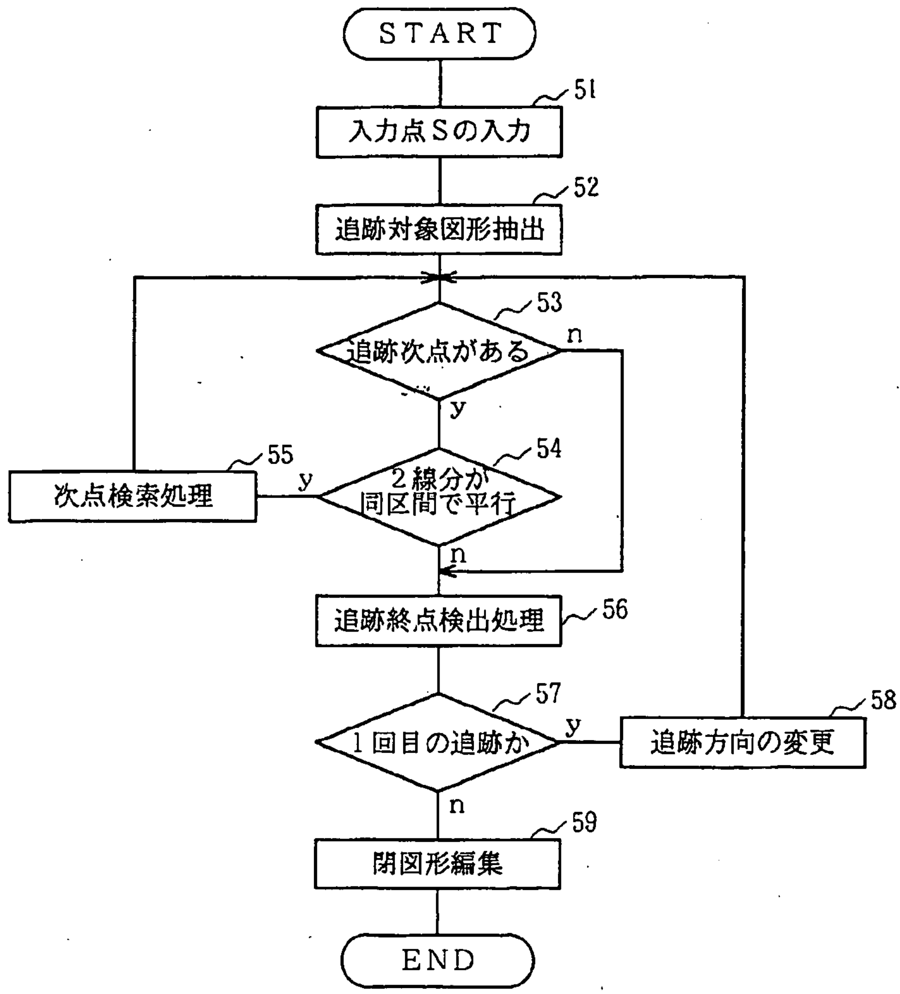
【図12】

図12



【図５】

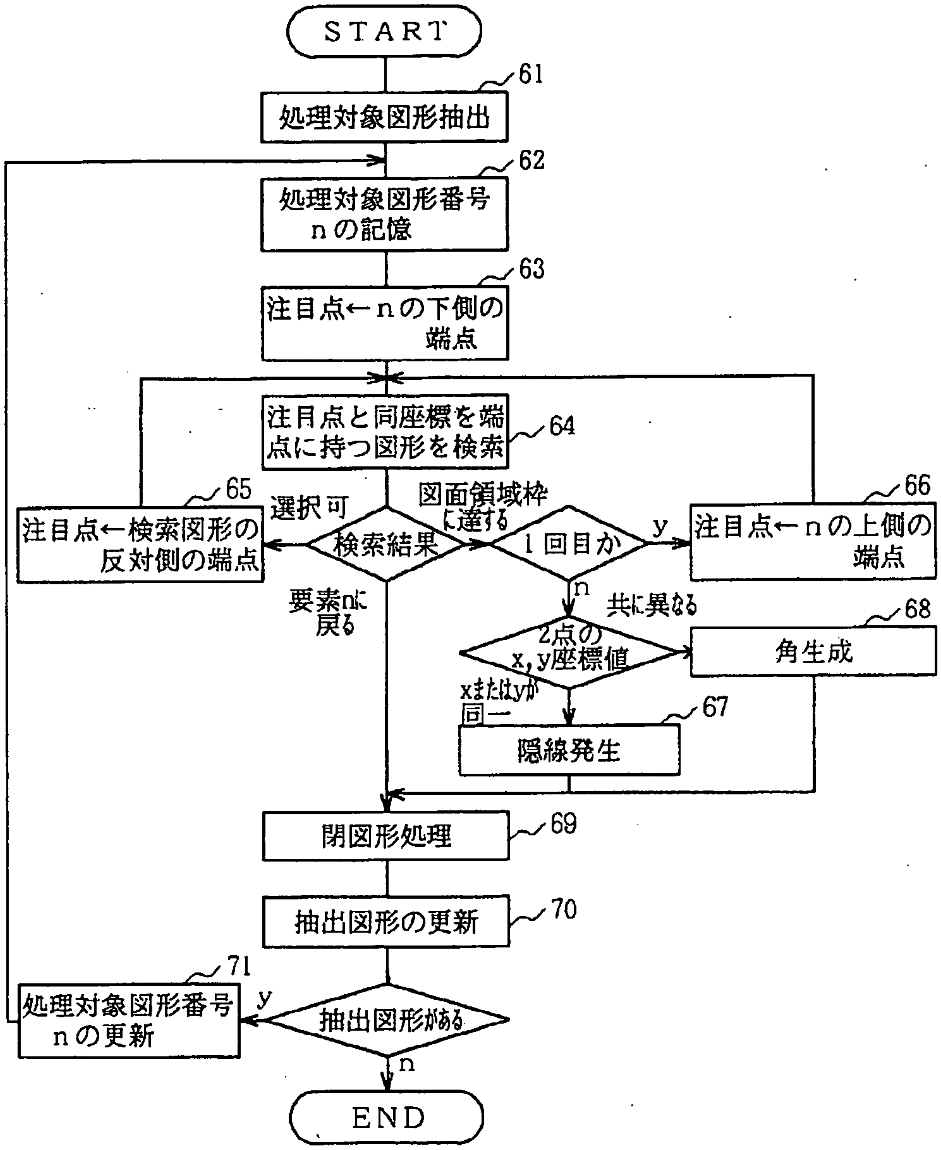
図５



START
51
入力点Ｓの入力
52
追跡対象図形抽出
53
追跡次点がある
n
y
54
2線分が同区間で平行
55
次点検索処理
y
n
追跡終点検出処理
56
57
1回目の追跡か
y
58
追跡方向の変更
n
59
閉図形編集
END

【図６】

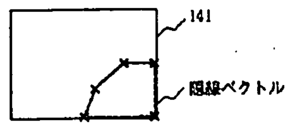
図６


START
61
処理対象図形抽出
62
処理対象図形番号
nの記憶
63
注目点←nの下側の
端点
注目点と同座標を端
点に持つ図形を検索
64
65
注目点←検索図形の
反対側の端点
選択可
検索結果
図面領域枠
に達する
1回目か
y
66
注目点←nの上側の
端点
要素nに
戻る
n
共に異なる
2点の
x, y座標値
68
角生成
xまたはyが
同一
67
隠線発生
閉図形処理
69
抽出図形の更新
70
71
処理対象図形番号
nの更新
y
抽出図形がある
n
END

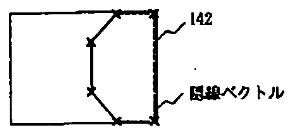
【図１４】

図１４

（１）　終点が、図面領域枠の隣合う辺上にある場合



（２）　終点が、図面領域枠の向かい合う辺上にある場合



フロントページの続き

(72)発明者　大場　雅浩
神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報システム事業部内

(72)発明者　末永　裕之
福岡県福岡市中央区渡辺通２丁目１番82号
九州電力株式会社内